A trade-off between classical and quantum circuit size for an attack against CSIDH

Jean-François Biasse, Xavier Bonnetain, Benjamin Pring, André Schrottenloher, William Youmans

A trade-off between classical and quantum circuit size for an attack against CSIDH

Jean-François Biasse[1], Xavier Bonnetain[2], Benjamin Pring[1], André Schrottenloher[2], William Youmans[1]

[1] University of South Florida, U.S.A.
[2] Inria, France

Abstract. We propose a heuristic algorithm to solve the underlying hard problem of the CSIDH cryptosystem (and other isogeny-based cryptosystems using elliptic curves with endomorphism ring isomorphic to an imaginary quadratic order $\mathcal{O}$). Let $\Delta = \mathrm{Disc}(\mathcal{O})$ (in CSIDH, $\Delta = -4p$ for p the security parameter). Let $0 < \alpha < 1/2$, our algorithm requires:
- A classical circuit of size $2^{\tilde{O}\left((\log|\Delta|)^{1-\alpha}\right)}$.
- A quantum circuit of size $2^{\tilde{O}\left((\log|\Delta|)^{\alpha}\right)}$.
- Polynomial classical and quantum memory.

Essentially, we propose to reduce the size of the quantum circuit below the state-of-the-art complexity $2^{\tilde{O}\left((\log|\Delta|)^{1/2}\right)}$ at the cost of increasing the classical circuit-size required. The required classical circuit remains subexponential, which is a superpolynomial improvement over the classical state-of-the-art exponential solutions to these problems. Our method requires polynomial memory, both classical and quantum.

Keywords: Isogenies, Imaginary quadratic orders, Quantum algorithms, Dihedral Hidden Subgroup Problem, CSIDH.

1 Introduction

Given two elliptic curves E_1, E_2 defined over a finite field $\mathbb{F}_q$, the isogeny problem consists in computing an isogeny $\phi : E_1 \to E_2$, i.e. a non-constant morphism that maps the identity point on E_1 to the identity point on E_2. A hash function construction based on supersingular isogeny graphs was first proposed in [9], with a security based on the hardness of computing isogenies. An isogeny-based key-exchange was described by Couveignes [12], and its concept was independently rediscovered by Stolbunov [31].

Childs, Jao and Soukharev observed in [10] that the problem of finding an isogeny between two ordinary elliptic curves E_1 and E_2 defined over $\mathbb{F}_q$ and having the same endomorphism ring could be reduced to the problem of solving the Hidden Subgroup Problem (HSP) for a generalized dihedral group. More specifically, if the endomorphism ring of the curves is isomorphic to an imaginary quadratic order $\mathcal{O}$, then the problem of finding an isogeny between E_1 and E_2 can be reduced to the problem of finding an ideal $\mathfrak{a} \subseteq \mathcal{O}$ such that $[\mathfrak{a}] * \overline{E}_1 = \overline{E}_2$

where $*$ is the action of the ideal class group $\mathrm{Cl}(\mathcal{O})$, $[\mathfrak{a}]$ is the class of $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O})$ and $\overline{E}_i$ is the isomorphism class of the curve E_i. Let $N := |\mathrm{Cl}(\mathcal{O})|$. Using Kuperberg's sieve [25], this task requires $2^{O\left(\sqrt{\log(N)}\right)}$ queries to an oracle that computes the action of the class of an element in $\mathrm{Cl}(\mathcal{O})$. Using the heuristic oracle of [4], the cost of the oracle can be brought down to $2^{\tilde{O}\left(\sqrt[3]{\log(N)}\right)}$, thus giving an overall complexity of $2^{O\left(\sqrt{\log(N)}\right)}$ where $N \approx \sqrt{|\Delta|}$.

Although neither the CRS [12, 31] nor the CSIDH (a similar system [8] using supersingular curves defined over $\mathbb{F}_p$) cryptosystems are NIST candidates, it is natural to evaluate their security according to the methodology proposed by NIST for its standardization process [26]. In particular, Level I is defined in [26, Page 16] as follows: "any attack that breaks [this] security definition must require computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit key (e.g. AES-128)." Hence, this corresponds to 2^{128} classical AES evaluations (2^{143} classical gates, according to the document) or to $2^{87.5}$ quantum gates (with 2953 logical qubits), according to the counts given in [17] on the universal Clifford + T set. We point out that this "or" has no reason to be exclusive: a quantum adversary can *also* run massive classical computations.

Contributions. We propose a different trade-off between classical and quantum circuits in the cryptanalysis of CRS and CSIDH relying on the resolution of the Hidden Shift Problem. Let E_1, E_2 be two elliptic curves and $\mathcal{O}$ be an imaginary quadratic order of discriminant Δ such that $\mathrm{End}(E_i) \simeq \mathcal{O}$ for $i = 1, 2$. Then assuming Heuristic 1 for constant $0 < \alpha < 1/2$ and Heuristic 2, there is a quantum algorithm for computing $[\mathfrak{a}]$ such that $[\mathfrak{a}] * \overline{E}_1 = \overline{E}_2$ requiring:

- A classical circuit of size $2^{\tilde{O}\left(\log(|\Delta|)^{1-\alpha}\right)}$.
- A quantum circuit of size $2^{\tilde{O}(\log(|\Delta|)^{\alpha})}$.
- Polynomial classical and quantum memory.

Related Works. After the publication of CSIDH, there has been a line of works on the quantum security of CRS and CSIDH. Some of these works concern the security of concrete CSIDH [8] parameters. These include [6] and [3], which give a quantum circuit for computing isogenies for the 512-bit CSIDH parameters. On the asymptotic side, which is our main focus here, both [4] and [19] present algorithms for computing isogenies with quantum (and classical) circuit size in $2^{\tilde{O}\left(\log(|\Delta|)^{1/2}\right)}$ and polynomial space, which yields a subexponential quantum attack on CSIDH and CRS with polynomial quantum space. While these two previous works focused on isogeny computations, in this paper, we complement the analysis of the Hidden Shift resolution underlying the attack procedure common to all these works. With our trade-off, we can obtain a superpolynomial improvement on the size of the quantum circuit.

The rest of the paper is organized as follows: Section 2 contains background information on isogenies. Section 3 shows the connection between the Dihedral Hidden Subgroup Problem and the computation of isogenies. Section 4 give a

high level description of the idea for the resolution of the Dihedral HSP. Section 5 introduces the concept of trading-off quantum gates for classical gates in the resolution of the Dihedral HSP. Section 6 Describes a heuristic oracle compatible with the intended trade-off. Section 7 discusses the heuristic made for the validity of the oracle. Section 8 describes the challenges of a fault-tolerant implementation. Section 9 concludes and discusses the relevance of this result to the evaluation of the security with respect to NIST security levels.

2 Mathematical background

An elliptic curve E defined over a finite field $\mathbb{F}_q$ of characteristic $p \neq 2, 3$ is a projective algebraic curve with an affine plane model given by an equation of the form $y^2 = x^3 + ax + b$, where a, $b \in \mathbb{F}_q$ and $4a^3 + 27b^2 \neq 0$. The set of points of an elliptic curve is equipped with an additive group law. Details about the arithmetic of elliptic curves can be found in many references, such as [30, Chap. 3].

Let E_1, E_2 be two elliptic curves defined over $\mathbb{F}_q$. An isogeny $\phi\colon E_1 \to E_2$ over $\mathbb{F}_q$ (resp. over $\overline{\mathbb{F}}_q$) is a non-constant rational map defined over $\mathbb{F}_q$ (resp. over $\overline{\mathbb{F}}_q$) which sends the identity point on E_1 to the identity point on E_2. The degree of an isogeny is its degree as a rational map, and an isogeny of degree ℓ is called an ℓ-isogeny. Moreover, E_1, E_2 are said to be isomorphic over $\mathbb{F}_q$, or $\mathbb{F}_q$-isomorphic, if there exist isogenies $\phi_1\colon E_1 \to E_2$ and $\phi_2\colon E_2 \to E_1$ over $\mathbb{F}_q$ whose composition is the identity. Two $\overline{\mathbb{F}}_q$-isomorphic elliptic curves have the same j-invariant given by $j := 1728\frac{4a^3}{4a^3+27b^2}$.

An order $\mathcal{O}$ in a number field K such that $[K : \mathbb{Q}] = n$ is a subring of K which is a $\mathbb{Z}$-module of rank n. A fractional ideal of $\mathcal{O}$, is a set of the form $\mathfrak{a} = \frac{1}{d}I$ where I is an ideal of $\mathcal{O}$ and $d \in \mathbb{Z}_{>0}$. A fractional ideal I is said to be invertible if there exists a fractional ideal J such that $IJ = \mathcal{O}$. The invertible fractional ideals form a multiplicative group $\mathcal{I}$. Let $\mathcal{P}$ be the subgroup consisting of the invertible principal ideals. The ideal class group $\mathrm{Cl}(\mathcal{O})$ is $\mathrm{Cl}(\mathcal{O}) := \mathcal{I}/\mathcal{P}$. We denote by $[\mathfrak{a}]$ the class of the fractional ideal $\mathfrak{a}$ in $\mathrm{Cl}(\mathcal{O})$. The ideal class group is finite and its cardinality $h_{\mathcal{O}}$ satisfies $h_{\mathcal{O}} \leq \sqrt{|\Delta|}\ln(|\Delta|)$ (see [11, §5.10.1]), where Δ is the discriminant of $\mathcal{O}$.

Let E be an elliptic curve defined over $\mathbb{F}_q$. An endomorphism of E is either an isogeny defined over $\overline{\mathbb{F}}_q$ between E and itself, or the zero morphism. The set of endomorphisms of E forms a ring that is denoted by $\mathrm{End}(E)$. For elliptic curves, $\mathrm{End}(E)$ is either an order in an imaginary quadratic field (and has $\mathbb{Z}$-rank 2) or a maximal order in a quaternion algebra ramified at p (the characteristic of the base field) and ∞ (and has $\mathbb{Z}$-rank 4). In the former case, E is said to be ordinary while in the latter it is called supersingular. When a supersingular curve is defined over $\mathbb{F}_p$, then the ring of its $\mathbb{F}_p$-endomorphisms, denoted by $\mathrm{End}_{\mathbb{F}_p}(E)$, is isomorphic to an imaginary quadratic order, much like in the ordinary case.

When E is ordinary (resp. supersingular over $\mathbb{F}_p$), the class group of $\mathrm{End}(E)$ (resp. $\mathrm{End}_{\mathbb{F}_p}(E)$) acts transitively on isomorphism classes of elliptic curves having the same endomorphism ring. More precisely, the class of an ideal $\mathfrak{a} \subseteq \mathcal{O}$ acts

on $\overline{E}$ with $\mathrm{End}(E) \simeq \mathcal{O}$ via an isogeny of degree $\mathcal{N}(\mathfrak{a})$ (the algebraic norm of $\mathfrak{a}$). Likewise, each isogeny $\varphi\colon E \to E'$ where $\mathrm{End}(E) \simeq \mathrm{End}(E') \simeq \mathcal{O}$ corresponds (up to isomorphism) to a class in $\mathrm{Cl}(\mathcal{O})$. From an ideal $\mathfrak{a}$ and the ℓ-torsion (where $\ell = \mathcal{N}(\mathfrak{a})$), one can recover the kernel of φ, and then using Vélu's formulae [34], one can derive the corresponding isogeny. We denote by $[\mathfrak{a}] * \overline{E}$ the action of the ideal class of $\mathfrak{a}$ on $\overline{E}$. To evaluate the action of $[\mathfrak{a}]$, we decompose it as a product of classes of prime ideals of small norm ℓ, and evaluate the action of each prime ideal as an ℓ-isogeny. This strategy was described by Couveignes [12], Galbraith–Hess–Smart [15], and later by Bröker–Charles–Lauter [7] and reused in many subsequent works.

3 Isogenies from solutions to the HSP

As shown in [5, 10], the computation of an isogeny between E_1 and E_2 defined over $\mathbb{F}_q$ such that there is an imaginary quadratic order with $\mathcal{O} \simeq \mathrm{End}(E_i)$ for $i = 1, 2$ can be done by exploiting the action of the ideal class group of $\mathcal{O}$ on isomorphism classes of curves with endomorphism ring isomorphic to $\mathcal{O}$. This concerns the cases of ordinary curves, and supersingular curves defined over $\mathbb{F}_p$.

Assume we are looking for $\mathfrak{a}$ such that $[\mathfrak{a}] * \overline{E}_1 = \overline{E}_2$. This is precisely the hard mathematical problem of the CSIDH [8] and CRS [12, 29] cryptosystems. Let $A = \mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_k} \simeq \mathrm{Cl}(\mathcal{O})$. We define $f : \mathbb{Z}_2 \ltimes A \to \mathbb{F}_q$ by

$$f(x, \boldsymbol{y}) := \begin{cases} [\mathfrak{a}_{\boldsymbol{y}}] * \overline{E}_1 & \text{if } x = 0, \\ [\mathfrak{a}_{-\boldsymbol{y}}] * \overline{E}_2 & \text{if } x = 1, \end{cases} \tag{1}$$

where $[\mathfrak{a}_{\boldsymbol{y}}]$ is the element of $\mathrm{Cl}(\mathcal{O})$ corresponding to $\boldsymbol{y} \in A$ via the isomorphism $\mathrm{Cl}(\mathcal{O}) \simeq A$. Let H be the subgroup of $\mathbb{Z}_2 \ltimes A$ such that $f(x, \boldsymbol{y}) = f(x', \boldsymbol{y}')$ if and only if $(x, \boldsymbol{y}) - (x', \boldsymbol{y}') \in H$. Then $H = \{(0, \boldsymbol{0}), (1, \boldsymbol{s})\}$ where $\boldsymbol{s} \in A$ such that $[\mathfrak{a}_{\boldsymbol{s}}] * \overline{E}_1 = \overline{E}_2$. The computation of $\boldsymbol{s}$ can thus be done through the resolution of the Hidden Subgroup Problem in $\mathbb{Z}_2 \ltimes A$.

Algorithm 1 Quantum algorithm for evaluating the action in $\mathrm{Cl}(\mathcal{O})$

Input: Elliptic curves E_1, E_2, imaginary quadratic order $\mathcal{O}$ such that $\mathrm{End}(E_i) \simeq \mathcal{O}$ for $i = 1, 2$ such that there is $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})$ satisfying $[\mathfrak{a}] * \overline{E}_1 = \overline{E}_2$.
Output: $[\mathfrak{a}]$
1: Compute $A = \mathbb{Z}_{d_1} \times \cdots \times \mathbb{Z}_{d_k}$ such that $A \simeq \mathrm{Cl}(\mathcal{O})$.
2: Find $H = \{(0, 0), (1, s)\}$ by solving the HSP in $\mathbb{Z}_2 \ltimes A$ with oracle (1).
3: **return** $[\mathfrak{a}_s]$

4 Sieve algorithms for solving the HSP

Kuperberg's original algorithm Assume that we want to find a secret subgroup $H = \{(0, 0), (1, d)\}$ in $D_N := \mathbb{Z}_2 \ltimes \mathbb{Z}_N$ given a function (oracle) $f : D_N \to$

X where X is a finite set. Additionally, we assume that $N = 2^n$ for simplicity. Using a circuit implementing f, we can prepare the state $|\psi_k^{d,N}\rangle := \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i \frac{kd}{N}}|1\rangle\right)$. We want to recover d from many states $|\psi_k^{d,N}\rangle$ where k is distributed uniformly at random in $\mathbb{Z}_N$. When we restrict ourselves to $N = 2^n$, this task consists in recovering d bit by bit. To get the least significant bit of d, we only need $|\psi_{2^{n-1}}^{d,2^n}\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + (-1)^d|1\rangle\right)$. As shown in [24], the repetition of this process yields all bits of d. When N is not a power of 2, the process is terminated with a quantum phase estimation step.

To go from many $|\psi_k^{d,N}\rangle$ with random k to $|\psi_{2^{n-1}}^{d,2^n}\rangle$, Kuperberg's sieve [24] proceeds by small iterations. Given two states $|\psi_{k_1}^{d,N}\rangle, |\psi_{k_2}^{d,N}\rangle$ where k_1, k_2 share the same initial l bits, there is a simple procedure that computes $|\psi_{k_1-k_2}^{d,N}\rangle$ with constant probability, thus killing l bits of the decomposition of the index k. At the end of the process we end up with states of the form $|\psi_{2^{n-1}}^{d,2^n}\rangle$ and $|\psi_0^{d,2^n}\rangle$. As we saw above, the latter gives us the least significant bit of d. The sieve starts with a set L_0 of states of the form $|\psi_k^{d,N}\rangle$ with $|L_0| = 2^{O(\sqrt{n})}$ and at each steps recombines all states sharing the same last $m = \lceil\sqrt{n-1}\rceil$ bits. At each step of the way, the cardinality of the set gets divided by 4. At the end, L_m contains states of the form $|\psi_{2^{n-1}}^{d,2^n}\rangle$ and $|\psi_0^{d,2^n}\rangle$. The cost of the procedure is dominated by the creation of L_0 with takes $2^{O(\sqrt{n})}$ calls to the circuit implementing f.

In CSIDH, $\mathrm{Cl}(\mathcal{O})$ is cyclic with high probability, but this applies to non-cyclic groups [10, Appendix A]. Here, we consider the HSP in D_N with $N = 2^n$.

Low memory variants The main disadvantage of Kuperberg's sieve is that the memory requirements are proportional to the gate complexity, which is in $2^{O(\sqrt{n})}$. That is a subexponential space complexity. Regev's variant [27] offers a classical and quantum polynomial space complexity at the cost of a slight increase of the runtime. The idea is to only keep a polynomial amount of qubits at all time and to recombine to produce states of the form $|\psi_k^{d,N}\rangle$ with initial bits of k being zero. Kuperberg also described a second Hidden Shift algorithm [25] that uses a different combination method. It has also a time cost in $2^{O(\sqrt{n})}$, and uses only a polynomial amount of qubits. It however has a classical memory requirement as large as the classical time.

5 Trade-off classical/quantum

Regev's variant of Kuperberg's sieve can be seen as an n_1-step process which is paused at each step to perform a classical brute-force enumeration of cost $2^{O(n_2)}$. Instead of balancing the classical and quantum effort, we propose to spend more effort performing the classical search to reduce the size of the quantum circuit. Let $n \approx n_1 n_2$, with $n_1 = O\left(n^\alpha\right)$ and $n_2 = O\left(n^{1-\alpha}\right)$ for some $0 < \alpha < 1$. The case $\alpha = 1/2$ is essentially Regev's variant [27].

Proposition 1. *Let $0 < \alpha < 1/2$, then there is a quantum algorithm to solve the HSP in D_N with a circuit satisfying:*

Algorithm 2 Iteration of the sieve procedure based on [27]

Input: Integers n_1, n_2 and n_2+4 states of the form $|\psi_{k_i}^{d,N}\rangle$ for random k_i having their initial tn_2 bits equal to 0.

Output: $|\psi_k^{d,N}\rangle$ for a random k having its initial $(t+1)n_2$ bits equal to 0.

1: $\boldsymbol{k} \leftarrow (k_1, \ldots, k_{n_2+4})$.

2: From $\bigotimes_{i \leq n_2+4} |\psi_{k_i}^{d,N}\rangle$, get $\frac{1}{\sqrt{2^{n_2+4}}} \sum_{\boldsymbol{b} \in \{0,1\}^{n_2+4}} e^{2i\pi d \frac{\langle \boldsymbol{b} \cdot \boldsymbol{k} \rangle}{N}} |\boldsymbol{b}\rangle |\langle \boldsymbol{b} \cdot \boldsymbol{k} \rangle \bmod 2^{n_2}\rangle$.

3: Measure the second register to obtain $z \in \{0, \ldots, 2^{n_2}-1\}$.

4: Compute the number m of $\boldsymbol{b} \in \{0,1\}^{n_2+4}$ such that $\langle \boldsymbol{b} \cdot \boldsymbol{k} \rangle \bmod 2^{n_2} = z$.

5: **if** $m \notin [2, 32]$ **then return** failure.

6: $\boldsymbol{b^1} \ldots, \boldsymbol{b^m} \leftarrow$ the m vectors that satisfy $\langle \boldsymbol{b^j} \cdot \boldsymbol{k} \rangle \bmod 2^{n_2} = z$.

7: $|\psi\rangle \leftarrow \frac{1}{\sqrt{2}} \left(|\boldsymbol{0}\rangle + e^{2i\pi d \frac{\langle (\boldsymbol{b^2} - \boldsymbol{b^1}) \cdot \boldsymbol{k} \rangle}{N}} |\boldsymbol{1}\rangle \right)$ with a measurement on Span $\left(\boldsymbol{b^1}, \boldsymbol{b^2}\right)$.

8: **return** $|\psi\rangle$.

- $2^{\tilde{O}(n^\alpha)}$ *calls to a circuit implementing* f *are made.*
- *The number of quantum gate beside the oracle is in* $2^{\tilde{O}(n^\alpha)}$.
- *The number of classical gates is in* $2^{O(n^{1-\alpha})}$.

Proof. As long as $n_2 \to \infty$, the main ingredients of the proof of the validity and run time of [27] still hold. Namely, a direct application of Chebyshev's inequality shows that Step 5 (and therefore Algorithm 2) has a constant probability of success. Following the approach of [27], the algorithm to solve the HSP consists in the production of states $|\psi_k^{d,N}\rangle$ for random k with an oracle implementing f, and 2^{n_1} successive applications of Algorithm 2 to produce $|\psi_{2^{n-1}}^{d,2^n}\rangle$. An application of the Chernoff bound shows that the number of calls to the oracle implementing f that guarantees the success of the overall procedure is $n_2^{O(n_1)} = 2^{\tilde{O}(n^\alpha)}$. Meanwhile, each brute force search of the number m of vectors $\boldsymbol{b} \in \{0,1\}^{n_2+4}$ such that $\langle \boldsymbol{b} \cdot \boldsymbol{k} \rangle \bmod 2^{n_2} = z$ is performed by a classical circuit of size $2^{O(n^{1-\alpha})}$.

The quality of the trade-off depends on the cost of the oracle. Indeed, if the quantum circuit to implement the oracle f is larger than $2^{\tilde{O}(n^\alpha)}$ for the chosen α, then the size of the circuit to implement f will dominate the number of quantum gates. This issue particularly impacts the resolution of the isogeny problem between elliptic curves whose endomorphism ring is isomorphic to an imaginary quadratic order (i.e. ordinary curves and supersingular curves defined over $\mathbb{F}_p$).

6 The cost of the isogeny oracle

Let $\mathfrak{p}_1, \ldots, \mathfrak{p}_u$ be prime ideals generating $\mathrm{Cl}(\mathcal{O})$. Let $\mathcal{L}$ be the lattice of relations between $\mathfrak{p}_1, \ldots, \mathfrak{p}_u$, i.e. the lattice of all the vectors $(f_1, \ldots, f_u) \in \mathbb{Z}^u$ such that $\prod_i \mathfrak{p}_i^{f_i}$ is principal. In other words, the ideal class $\left[\prod_i \mathfrak{p}_i^{f_i}\right]$ is the neutral element of $\mathrm{Cl}(\mathcal{O})$. The high-level strategy for computing the action of $[\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})$ on $\overline{E_1}$

is the following: (i) Compute a basis B for $\mathcal{L}$, (ii) Find a BKZ-reduced basis B' of $\mathcal{L}$, (iii) Find $(h_1, \ldots, h_u) \in \mathbb{Z}^u$ such that $[\mathfrak{a}] = \left[\prod_i \mathfrak{p}_i^{h_i}\right]$, (iv) Use Babai's nearest plane method on B' to find short $(h'_1, \ldots, h'_u) \in \mathbb{Z}^u$ such that $[\mathfrak{a}] = \left[\prod_i \mathfrak{p}_i^{h'_i}\right]$, (v) Evaluate the action of $\left[\prod_i \mathfrak{p}_i^{h'_i}\right]$ on $\overline{E}_1$ by applying repeatedly the action of the $\mathfrak{p}_i$ for $i = 1, \ldots, u$. Step 1 is a precomputation. It takes quantum polynomial time. Step 2 can be performed as a precomputation requiring only classical gates.

Heuristic 1 (With parameter $0 < \alpha < 1/2$) *Let $0 < \alpha < 1/2$ and $\mathcal{O}$ be an imaginary quadratic order of discriminant Δ. There are $(\mathfrak{p}_i)_{i \leq k}$ for $k = \log^{1-\alpha}(|\Delta|)$ split prime ideals of norm in* $\mathrm{Poly}(\log(|\Delta|)$ *whose classes generate* $\mathrm{Cl}(\mathcal{O})$. *Furthermore, each class of* $\mathrm{Cl}(\mathcal{O})$ *has a representative of the form $\prod_i \mathfrak{p}_i^{n_i}$ for $|n_i| \leq e^{\log^{\alpha}|\Delta|}$.*

Algorithm 3 Precomputation for the oracle

Input: Order $\mathcal{O}$ of discriminant Δ and $0 < \alpha < 1/2$.

Output: Split prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$ whose classes generate $\mathrm{Cl}(\mathcal{O})$ where $s = \log^{1-\alpha}(|\Delta|)$, reduced basis B' of the lattice $\mathcal{L}$ of vectors $(e_1, \ldots, e_s)$ such that $\left[\prod_i \mathfrak{p}_i^{e_i}\right]$ is trivial, generators $\mathfrak{g}_1, \ldots, \mathfrak{g}_l$ such that $\mathrm{Cl}(\mathcal{O}) = \langle \mathfrak{g}_1 \rangle \times \cdots \times \langle \mathfrak{g}_l \rangle$ and vectors $\boldsymbol{v}_i$ such that $\mathfrak{g}_i = \prod_j \mathfrak{p}_j^{v_{i,j}}$.

1: Find $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$ satisfying the conditions of Heuristic 1 with [4, Alg. 2].
2: $\mathcal{L} \leftarrow$ lattice of vectors $(e_1, \ldots, e_s)$ such that $\prod_i \mathfrak{p}_i^{e_i}$ is principal.
3: Compute a BKZ-reduced matrix $B' \in \mathbb{Z}^{s \times s}$ of a basis of $\mathcal{L}$ with block size $\log^{1-2\alpha}(|\Delta|)$.
4: Compute $U, V \in \mathrm{GL}_s(\mathbb{Z})$ such that $UB'V = \mathrm{diag}(d_1, \ldots, d_s)$ is the Smith Normal Form of B'.
5: $l \leftarrow \min_{i \leq s}\{i \mid d_i \neq 1\}$. For $i \leq l$, $\boldsymbol{v_i} \leftarrow i$-th column of V.
6: $V' \leftarrow V^{-1}$. For $i \leq l$, $\mathfrak{g}_i \leftarrow \prod_{j \leq s} \mathfrak{p}_j^{v'_{i,j}}$.
7: **return** $\{\mathfrak{p}_1, \ldots, \mathfrak{p}_s\}$, B', $\{\mathfrak{g}_1, \ldots, \mathfrak{g}_l\}$, $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_l\}$.

Lemma 1. *Let $\mathcal{L}$ be an n-dimensional lattice with input basis $B \in \mathbb{Z}^{n \times n}$, and let $\beta < n$ be a block size. Then the BKZ variant of [18] used with Kannan's enumeration technique [22] returns a basis $\boldsymbol{b}'_1, \ldots, \boldsymbol{b}'_n$ such that $\|\boldsymbol{b}'_1\| \leq e^{\frac{n}{\beta}\ln(\beta)(1+o(1))}\lambda_1(\mathcal{L})$, using time* $\mathrm{Poly}(n, \mathrm{Size}(B))\beta^{\beta\left(\frac{1}{2e}+o(1)\right)}$ *and polynomial space.*

Proof. See proof of [4, Lem. 1]

Corollary 1. *Assuming Heuristic 1 for α, Algorithm 3 is correct, runs in time $2^{\tilde{O}\left(\log(|\Delta|)^{1-2\alpha}\right)}$ and has polynomial space complexity. It returns a basis of $\mathcal{L}$ whose first vector $\boldsymbol{b}'_1$ satisfies $\|\boldsymbol{b}'_1\| \leq 2^{\tilde{O}(\log(|\Delta|)^{\alpha})}$.*

We implement Algorithm 4 reversibly by using generic techniques due to Bennett [2] to convert any algorithm taking time T and space S into a reversible

Algorithm 4 Quantum oracle for implementing f defined in (1)

Input: Curves E_1, E_2. Order $\mathcal{O}$ of discriminant Δ such that $\mathrm{End}(E_i) \simeq \mathcal{O}$ for $i = 1, 2$. Split prime ideals $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$ whose classes generate $\mathrm{Cl}(\mathcal{O})$ where $s = \log^{1-\alpha}(|\Delta|)$, reduced basis B' of the lattice $\mathcal{L}$ of vectors $(e_1, \ldots, e_s)$ such that $\left[\prod_i \mathfrak{p}_i^{e_i}\right]$ is trivial, generators $\mathfrak{g}_1, \ldots, \mathfrak{g}_l$ such that $\mathrm{Cl}(\mathcal{O}) = \langle \mathfrak{g}_1 \rangle \times \cdots \times \langle \mathfrak{g}_l \rangle$ and vectors $\boldsymbol{v}_i$ such that $\mathfrak{g}_i = \prod_j \mathfrak{p}_j^{v_{i,j}}$. Ideal class $[\mathfrak{a}_{\boldsymbol{y}}] \in \mathrm{Cl}(\mathcal{O})$ represented by the vector $\boldsymbol{y} = (y_1, \ldots, y_l) \in \mathbb{Z}/d_1\mathbb{Z} \times \cdots \times \mathbb{Z}/d_l\mathbb{Z} \simeq \mathrm{Cl}(\mathcal{O})$, and $x \in \mathbb{Z}/2\mathbb{Z}$.

Output: $f(x, \boldsymbol{y})$.

1: $\boldsymbol{y} \leftarrow \sum_{i \leq l} y_i \boldsymbol{v_i} \in \mathbb{Z}^s$ (now $[\mathfrak{a}_{\boldsymbol{y}}] = \left[\prod_i \mathfrak{p}_i^{y_i}\right]$).
2: Use Babai's nearest plane method with the basis B' to find $\boldsymbol{u} \in \mathcal{L}$ close to $\boldsymbol{y}$.
3: $\boldsymbol{y} \leftarrow \boldsymbol{y} - \boldsymbol{u}$.
4: **If** $x = 0$ **then** $\overline{E} \leftarrow \overline{E}_1$ **else** $\overline{E} \leftarrow \overline{E}_2$.
5: **for** $i \leq s$ **do**
6: **for** $j \leq y_i$ **do**
7: $\overline{E} \leftarrow [\mathfrak{p}_i] * \overline{E}$.
8: **end for**
9: **end for**
10: **return** $|\overline{E}\rangle$.

algorithm taking time $T^{1+\epsilon}$, for an arbitrary small $\epsilon > 0$, and space $O(S \log T)$. To bound the cost of Algorithm 4, we assume the following standard heuristic.

Heuristic 2 (GSA) *The basis B' computed in Algorithm 3 satisfies the Geometric Series Assumption (GSA): there is $0 < q < 1$ such that $\|\widehat{\boldsymbol{b}'_{\boldsymbol{i}}}\| = q^{i-1}\|\boldsymbol{b_1}\|$ where $\left(\widehat{\boldsymbol{b}'_{\boldsymbol{i}}}\right)_{i \leq n}$ is the Gram-Schmidt basis corresponding to B'.*

Proposition 2. *Assuming Heuristic 1 for $0 < \alpha < 1/2$ and Heuristic 2, Algorithm 4 is correct and runs in quantum time $2^{\tilde{O}(\log(|\Delta|)^\alpha)}$ with polynomial space.*

Proof. Each group action of Step 7 is polynomial in $\log(p)$ and in $\mathcal{N}(\mathfrak{p}_i)$. Moreover, Babai's algorithm runs in polynomial time and returns $\boldsymbol{u}$ such that

$$\|\boldsymbol{y} - \boldsymbol{u}\| \leq \frac{1}{2}\sqrt{\sum_i \|\widehat{\boldsymbol{b}'_{\boldsymbol{i}}}\|^2} \leq \frac{1}{2}\sqrt{n}\|\boldsymbol{b'}_1\| \in 2^{\tilde{O}(\log(|\Delta|)^\alpha)}.$$

Therefore, the y_i are in $2^{\tilde{O}(\log(|\Delta|)^\alpha)}$, which is the cost of Steps 5 to 9. The main observation allowing us to reduce the search to a close vector to the computation of a BKZ-reduced basis is that Heuristic 1 gives us the promise that there is $\boldsymbol{u} \in \mathcal{L}$ at distance less than $2^{\tilde{O}(\log(|\Delta|)^\alpha)}$ from $\boldsymbol{y}$.

Corollary 2. *Let E_1, E_2 be two elliptic curves and $\mathcal{O}$ be an imaginary quadratic order of discriminant Δ such that $\mathrm{End}(E_i) \simeq \mathcal{O}$ for $i = 1, 2$. Then assuming Heuristic 1 for $0 < \alpha < 1/2$, Algorithm 5 finds $[\mathfrak{a}]$, with $[\mathfrak{a}] * \overline{E}_1 = \overline{E}_2$ using:*

- *A classical circuit of size $2^{\tilde{O}\left(\log(|\Delta|)^{1-\alpha}\right)}$.*

Algorithm 5 Hybrid algorithm for finding the group action.

Input: Curves E_1, E_2, $0 < \alpha < 1/2$, order $\mathcal{O}$ such that $\mathrm{End}(E_i) \simeq \mathcal{O}$ for $i = 1, 2$, n_1, n_2 with $N = 2^{1+n_1 n_2}$ for $\mathrm{Cl}(\mathcal{O}) \simeq \mathbb{Z}_N$.
Output: $X \in \mathbb{Z}_N \leftrightarrow [\mathfrak{a}] \in \mathrm{Cl}(\mathcal{O})$ such that $[\mathfrak{a}] * \overline{E_1} = \overline{E_2}$.

1: Compute $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$, B', $\mathfrak{g}$, $\boldsymbol{v}$ with Algorithm 3.
2: $b \leftarrow 0$, $n \leftarrow 0$, $X \leftarrow 0$, f_n defined by (1).
3: **while** $n < 1 + n_1 n_2$ **do**
4: Repeat Algorithm 5 using the oracle f_n implemented with Algorithm 4 and using $\mathfrak{p}_1, \ldots, \mathfrak{p}_s$, B', $\mathfrak{g}$, $\boldsymbol{v}$ to compute $b \in \{0, 1\}$.
5: $X \leftarrow X + b2^n$, $n \leftarrow n + 1$, $f_n \leftarrow \{(x, y) \in D_{N/2^n} \mapsto f_{n-1}(x, 2y + b)\}$.
6: **end while**
7: **return** X.

– *A quantum circuit of size* $2^{\tilde{O}(\log(|\Delta|)^\alpha)}$.
– *Polynomial classical and quantum memory.*

Similar modifications to [24] and [10, Appendix A] extend this to arbitrary class groups.

7 Discussion on Heuristic 1

The idea behind Heuristic 1 is that the number of vectors of length $\log(|\Delta|)^{1-\alpha}$ with entries bounded by $e^{\log(|\Delta|)^\alpha}$ is $|\Delta|$ while $|\mathrm{Cl}(\mathcal{O})| \approx \sqrt{|\Delta|}$. If the class of $\prod_i \mathfrak{p}_i^{x_i}$ yielded by a vector $\boldsymbol{x}$ were known to be distributed uniformly at random in $\mathrm{Cl}(\mathcal{O})$, then we would cover all of $\mathrm{Cl}(\mathcal{O})$ with high probability. Unfortunately, the distribution of the classes of these ideals is not known (unless we consider products over the first $\log(|\Delta|)^{2+\varepsilon}$ split primes [20], but this is incompatible with our restriction on α). To support Heuristic 1, we drew 5000 elements of $\mathrm{Cl}(\mathcal{O})$ for various $\mathcal{O}$ of increasing discriminant. At each discriminant size, we report the maximal exponent in the decomposition of the random classes with respect to the fist $\log(|\Delta|)^{1-\alpha}$ split primes. We systematically observe that it is significantly lower than $e^{\log(|\Delta|)^\alpha}$. In Table 1, we present the evolution of the maximal exponent for $\alpha = 0.4$ and $\mathrm{Disc}(\mathcal{O}) = -p$ for p the first prime greater than 2^i such that $-p$ is a fundamental discriminant and i between 35 and 160. In Appendix A we present similar results for $\alpha = 0.1, \ldots, 0.5$ and smaller increments in the size of Δ. Heuristic 1 intersects ongoing research in number theory, and it is a motivation for more study on the structure of the class group. The samples presented in this paper are admittedly low, but they support the fact that Heuristic 1 holds true more than 98% of the time (at least for the sizes of Δ that were inspected). Such a success rate makes Heuristic 1 relevant for discussions within the field of cryptography.

8 On fault tolerant implementations

All the asymptotic results regarding the proposed trade-off between classical and quantum circuits only apply to logical qubits. If we incorporate the cost of error

Table 1. Maximal exponent in short decompositions (over 5000 random elements of the class group).

$\log_2(\lvert\Delta\rvert)$	$\log^{0.6}(\lvert\Delta\rvert)$	Maximal exponent	$e^{\log^{0.4}(\lvert\Delta\rvert)}$
35	7	4	36
60	9	8	85
85	12	11	165
110	13	19	287
135	15	24	466
160	17	30	718

correction, then the quantum circuit has to idle while the classical circuit searches for the number m of vectors $\boldsymbol{b} \in \{0,1\}^{n_2+4}$ such that $\langle \boldsymbol{b} \cdot \boldsymbol{k} \rangle \bmod 2^{n_2} = z$. The logical gate representation of this circuit does not include the cost of idling, but in all realistic models of fault tolerant qubits, operations need to be performed on a qubit that is being stored while the classical computation is being done. There is currently an ongoing debate in the cryptographic community as to how to assign a cost-metric to a quantum algorithm given its representation in the logical quantum circuit-model of computation [3, 21]. One approach is the quantum circuit-size and the other is the product of the quantum circuit-width (#qubits) and the quantum circuit-depth (time taken). We have previously studied our tradeoff in light of the circuit-size metric. We now briefly make some remarks with regards to the latter, which is proposed as it captures the difficulties in performing quantum error-correction.

Regardless of the architecture chosen for quantum computers and method used to perform quantum error-correction, it is clear from theoretical error models regarding *physical* qubits that if we consider discrete timesteps, then applying single or two-qubit gates induce an error in the qubit with a significantly higher probability than if it were simply resting (or "idling") [23, 28, 13, 33, 14]. As the resources we must expend on error-correction is intrinsically linked to the probability of an error occuring, it is plain that the resources to protect an idle quantum state have the potential to be lower than those required to protect a quantum state undergoing active manipulation. For one example of the proposed gaps and tradeoffs that can exist for different architectures, see [32, Tab 2]. In Table 2, we observe that the error rate while storing a qubit is lower than when applying gates in most system.

Table 2. Gates and Memory Errors (Table 3 of [32]).

Error	Superconductors	Ion Traps	Quantum Dots	Photonics I
Gate	1.00×10^{-5}	3.19×10^{-9}	9.89×10^{-1}	1.01×10^{-1}
Memory	1.00×10^{-5}	2.52×10^{-12}	3.47×10^{-2}	9.80×10^{-4}

Furthermore, classical gates could be significantly faster in practice than quantum gates, thus reducing the quantum cost of idling. In fact, most recent resource estimations [1] can show that, given the current trajectory of quantum architectures, a quantum computation requires inherently a corresponding amount of classical computations. From the counts in [16] a Grover search for an AES-128 key requires 2^{106} classical computations, hence approximately 2^{20} classical computations per quantum gate.

Our tradeoff therefore allows for agility in cryptanalysis depending upon the eventual architecture of quantum computers and opens the door for improvements and further tradeoffs if smarter methods of performing the brute-force enumeration step are discovered. A simple example of a further trade-off would be to employ parallelism in this stage so that if m classical processors are available, then the classical time would be proportional to $2^{O(n^{1-\alpha})}/m + O(m)$, thus reducing the time of quantum idling even more. A full examination of this work under current projections involving quantum error-correction is left for future work.

9 Conclusion

We proposed an asymptotic trade-off between the size of the classical and quantum circuits required to attack CSIDH. This angle is motivated by the fact that to use the full power of the NIST metric, we should authorize 2^{128} classical computations and $2^{87.5}$ quantum gates *simultaneously.* This work showed that such a hybrid attack could be performed with a quantum and a classical circuit that are both asymptotically smaller than the state-of-the-art. The study of the impact of this attack against the parameters for a specific security level (ex: Level I) is left for future work. In the case of CSIDH-512, the number of Clifford + T gates required to run a reversible CSIDH isogeny computation has been estimated in [3] to approximately 2^{51}. This is costly, but if we adjust α such that $\log(|\Delta|)^{1-\alpha} \approx 128$ for $\log(|\Delta|) = 512$ (since $\log(|\Delta|) \approx \log(p)$ where p is the security parameter), we get $\alpha \approx 0.22$. Then $\log(|\Delta|)^{\alpha} \approx 4$, which indicates that the size of the quantum circuit besides oracle calls might be moderate, thus leaving the door open for the relevance of our algorithms to the analysis of the NIST Level I security of CSIDH.

Acknowledgments This work was supported by the U.S. National Science Foundation under grant 1839805, by NIST under grant 60NANB17D184, by a Seed Grant of the Florida Center for Cybersecurity, by the USF Proposal Enhancement Grant, and by the ERC Starting Grant QUASYModo.

References

1. M. Amy, O. Di Matteo, V. Gheorghiu, M. Mosca, A. Parent and J. Schanck, Estimating the Cost of Generic Quantum Pre-image Attacks on SHA-2 and SHA-3, in: *SAC*, Lecture Notes in Computer Science 10532, pp. 317–337, Springer, 2016.

2. C. H. Bennett, Time/space trade-offs for reversible computation, *SIAM Journal on Computing* **18** (1989), 766–776.
3. D. Bernstein, T. Lange, C. Martindale and L. Panny, Quantum Circuits for the CSIDH: Optimizing Quantum Evaluation of Isogenies, in: *EUROCRYPT (2)*, Lecture Notes in Computer Science 11477, pp. 409–441, Springer, 2019.
4. J.-F. Biasse, A. Iezzi and M. Jacobson Jr., A Note on the Security of CSIDH, in: *Progress in Cryptology - INDOCRYPT 2018 - 19th International Conference on Cryptology in India, New Delhi, India, December 9-12, 2018, Proceedings* (D. Chakraborty and T. Iwata, eds.), Lecture Notes in Computer Science 11356, pp. 153–168, Springer, 2018.
5. J.-F. Biasse, D. Jao and A. Sankar, A Quantum Algorithm for Computing Isogenies between Supersingular Elliptic Curves, in: *Progress in Cryptology - INDOCRYPT 2014 - 15th International Conference on Cryptology in India, New Delhi, India, December 14-17, 2014, Proceedings* (W. Meier and D. Mukhopadhyay, eds.), Lecture Notes in Computer Science 8885, pp. 428–442, Springer, 2014.
6. X. Bonnetain and A. Schrottenloher, *Quantum Security Analysis of CSIDH and Ordinary Isogeny-based Schemes*, Cryptology ePrint Archive, Report 2018/537, 2018, `https://eprint.iacr.org/2018/537`.
7. R. Bröker, D. Xavier Charles and K. Lauter, Evaluating Large Degree Isogenies and Applications to Pairing Based Cryptography, in: *Pairing-Based Cryptography - Pairing 2008, Second International Conference, Egham, UK, September 1-3, 2008. Proceedings* (S. Galbraith and K. Paterson, eds.), Lecture Notes in Computer Science, pp. 100–112, Springer, 2008.
8. W. Castryck, T. Lange, C. Martindale, L. Panny and J. Renes, *CSIDH: An Efficient Post-Quantum Commutative Group Action*, Cryptology ePrint Archive, Report 2018/383, 2018, `https://eprint.iacr.org/2018/383`, to appear in Asiacrypt 2018.
9. D. Charles, K. Lauter and E. Goren, Cryptographic hash functions from expander graphs, *Jornal of cryptology* **22** (2009), 93–113.
10. A. Childs, D. Jao and V. Soukharev, Constructing elliptic curve isogenies in quantum subexponential time, *Journal of Mathematical Cryptology* **8** (2013), 1 – 29.
11. H. Cohen, *A course in computational algebraic number theory*, Graduate Texts in Mathematics 138, Springer-Verlag, 1991.
12. J.-M. Couveignes, *Hard homgeneous spaces*, http://eprint.iacr.org/2006/291.
13. D. Crow, R. Joynt and M. Saffman, Improved error thresholds for measurement-free error correction, *Physical review letters* **117** (2016), 130503.
14. A. Fowler, M. Mariantoni, J. Martinis and A. Cleland, Surface codes: Towards practical large-scale quantum computation, *Physical Review A* **86** (2012), 032324.
15. S. Galbraith, F. Hess and N. Smart, Extending the GHS Weil Descent Attack, in: *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings* (L. Knudsen, ed.), Lecture Notes in Computer Science 2332, pp. 29–44, Springer, 2002.
16. V. Gheorghiu and M. Mosca, Quantum cryptanalysis of symmetric, public-key and hash-based cryptographic schemes, *arXiv preprint arXiv:1902.02332* (2019).
17. M. Grassl, B. Langenberg, M. Roetteler and R. Steinwandt, Applying Grover's Algorithm to AES: Quantum Resource Estimates, in: *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings* (T. Takagi, ed.), Lecture Notes in Computer Science 9606, pp. 29–43, Springer, 2016.

18. G. Hanrot, X. Pujol and D. Stehlé, Terminating BKZ, *IACR Cryptology ePrint Archive* **2011** (2011), 198.
19. D. Jao, J. LeGrow, C. Leonardi and L. Ruiz-Lopez, *A Subexponential-Time, Polynomial Quantum Space Algorithm for Inverting the CM Action*, Slides of presentation at the MathCrypt conference, 2018, `https://drive.google.com/file/d/15nkb9j0GKyLujYfAb8Sfz3TjBY5PWOCT/view`.
20. D. Jao, D. Miller, S. and R. Venkatesan, Expander graphs based on GRH with an application to elliptic curve cryptography, *J. Number Theory* **129** (2009), 1491–1504.
21. S. Jaques and J. Schanck, Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE, *IACR Cryptology ePrint Archive* **2019** (2019), 103, To appear in the proceedings of CRYPTO 2019.
22. R. Kannan, Improved Algorithms for Integer Programming and Related Lattice Problems, in: *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA* (D. Johnson, S. Fagin, M. Fredman, D. Harel, R. Karp, N. Lynch, C. Papadimitriou, R. Rivest, W. Ruzzo and J. Seiferas, eds.), pp. 193–206, ACM, 1983.
23. E. Knill, R. Laflamme and W. Zurek, Resilient quantum computation: error models and thresholds, *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* **454** (1998), 365–384.
24. G. Kuperberg, A Subexponential-Time Quantum Algorithm for the Dihedral Hidden Subgroup Problem, *SIAM J. Comput.* **35** (2005), 170–188.
25. G. Kuperberg, Another Subexponential-time Quantum Algorithm for the Dihedral Hidden Subgroup Problem, in: *8th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2013, May 21-23, 2013, Guelph, Canada* (S. Severini and F. Brandão, eds.), LIPIcs 22, pp. 20–34, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.
26. NIST, *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process*, 2016.
27. O. Regev, *A Subexponential Time Algorithm for the Dihedral Hidden Subgroup Problem with Polynomial Space*, arXiv:quant-ph/0406151.
28. B. Reichardt, Fault-tolerant quantum error correction for Steane's seven-qubit color code with few or no extra qubits, *arXiv preprint arXiv: 1804.06995* (2018).
29. A. Rostovtsev and A. Stolbunov, Public-Key Cryptosystem Based on Isogenies, *IACR Cryptology ePrint Archive* **2006** (2006), 145.
30. J. Silverman, *The arithmetic of elliptic curves*, Graduate texts in Mathematics 106, Springer-Verlag, 1992.
31. A. Stolbunov, Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves, *Adv. in Math. of Comm.* **4** (2010), 215–235.
32. M. Suchara, A. Faruque, C.-Y. Lai, G. Paz, F. Chong and J. Kubiatowicz, *Estimating the Resources for Quantum Computation with the QuRE Toolbox*, EECS Department, University of California, Berkeley, Report no. UCB/EECS-2013-119, May 2013, `http://www2.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-119.html`.
33. K. Svore, B. Terhal and D. DiVincenzo, Local fault-tolerant quantum computation, *Physical Review A* **72** (2005), 022317.
34. J. Vélu, Isogénies entre courbes elliptiques, *C. R. Acad. Sci. Paris Sér. A-B* **273** (1971), A238–A241.

A Numerical data in support of Heuristic 1

In this section, we provide additional numerical data in support of the heuristic made in Section 7. For each i in $30, 35, \ldots, 160$ and $\alpha = 0.1, \ldots, 0.5$, we select the first prime $p \geq 2^i$ such that $\Delta = -p$ is a fundamental discriminant. For each discriminant, we compute the corresponding ideal class group and produce a reduced basis of the lattice of relations between the classes of the split primes $\mathfrak{p}_i$ of norm less than $\log^{1-\alpha}(|\Delta|)$. Then we draw 5000 ideal classes uniformly at random and compute a short decomposition over the split primes of norm less than $\log^{1-\alpha}(|\Delta|)$. To compute a short decomposition of $[\mathfrak{a}]$, we solve an instance of the approximate Closest Vector Problem between a vector $\boldsymbol{x}$ such that $[\prod_i \mathfrak{p}_i] = [\mathfrak{a}]$ and the lattice $\mathcal{L}$ of relations. We solve approximate CVP by reducing the basis of $\mathcal{L}$ with the BKZ algorithm and calling Babai's nearest plane algorithm. We do not necessarily find the shortest $\boldsymbol{x}$, however, all our exponents are below the intended bound $e^{\log^{\alpha}(|\Delta|)}$. In each table, we show the largest exponent occurring in a decomposition next to $e^{\log^{\alpha}(|\Delta|)}$ for each Δ. Our heuristic is systematically satisfied. Moreover, aside from the case $\alpha = 0.1$ where the intended bound is already very small (between 4 and 5), we observe that our heuristic seems in fact very conservative. For example, for $\log_2(|\Delta|) = 160$ and $\alpha = 0.5$, the maximal exponent recorded over 5000 short decompositions is 188 while the intended bound is $e^{\log^{0.5}(|\Delta|)} = 37462$.

Table 3. Maximal exponent in short decompositions (over 5000 random elements of the class group).

$\log_2(\lvert\Delta\rvert)$	$\log^{0.9}(\lvert\Delta\rvert)$	Maximal exponent	$e^{\log^{0.1}(\lvert\Delta\rvert)}$
30	15	2	4
35	18	2	4
40	20	2	4
45	22	2	4
50	24	2	4
55	26	2	4
60	29	2	4
65	31	2	4
70	33	3	4
75	35	3	4
80	37	3	4
85	39	3	4
90	41	3	5
95	43	3	5
100	45	3	5
105	47	3	5
110	49	3	5
115	51	3	5
120	53	3	5
125	55	3	5
130	57	3	5
135	59	4	5
140	61	3	5
145	63	3	5
150	65	3	5
155	67	3	5
160	69	4	5

Table 4. Maximal exponent in short decompositions (over 5000 random elements of the class group).

$\log_2(\lvert\Delta\rvert)$	$\log^{0.8}(\lvert\Delta\rvert)$	Maximal exponent	$e^{\log^{0.2}(\lvert\Delta\rvert)}$
30	11	2	6
35	13	2	7
40	14	3	7
45	16	2	7
50	17	3	8
55	18	3	8
60	20	3	8
65	21	3	9
70	22	3	9
75	24	3	9
80	25	3	9
85	26	3	10
90	27	4	10
95	28	4	10
100	30	4	10
105	31	4	11
110	32	5	11
115	33	4	11
120	34	4	11
125	35	4	11
130	37	4	12
135	38	4	12
140	39	5	12
145	40	5	12
150	41	5	13
155	42	5	13
160	43	5	13

Table 5. Maximal exponent in short decompositions (over 5000 random elements of the class group).

$\log_2(\lvert\Delta\rvert)$	$\log^{0.7}(\lvert\Delta\rvert)$	Maximal exponent	$e^{\log^{0.3}(\lvert\Delta\rvert)}$
30	8	2	12
35	9	3	14
40	10	4	15
45	11	3	17
50	12	5	18
55	13	4	20
60	14	4	21
65	14	5	23
70	15	6	25
75	16	5	26
80	17	6	28
85	17	6	30
90	18	7	32
95	19	6	34
100	19	6	35
105	20	7	37
110	21	7	39
115	21	8	41
120	22	7	43
125	23	7	45
130	23	8	47
135	24	8	50
140	25	8	52
145	25	9	54
150	26	9	56
155	26	9	58
160	27	10	61

Table 6. Maximal exponent in short decompositions (over 5000 random elements of the class group).

$\log_2(\lvert\Delta\rvert)$	$\log^{0.6}(\lvert\Delta\rvert)$	Maximal exponent	$e^{\log^{0.4}(\lvert\Delta\rvert)}$
30	6	3	29
35	7	4	36
40	7	7	44
45	8	6	52
50	8	9	62
55	9	8	73
60	9	8	85
65	10	7	98
70	10	11	113
75	11	10	129
80	11	12	146
85	12	11	165
90	12	14	186
95	12	16	208
100	13	14	233
105	13	18	259
110	13	19	287
115	14	17	318
120	14	20	351
125	15	18	387
130	15	22	425
135	15	24	466
140	16	22	510
145	16	24	557
150	16	25	607
155	17	26	661
160	17	30	718

Table 7. Maximal exponent in short decompositions (over 5000 random elements of the class group).

$\log_2(\lvert\Delta\rvert)$	$\log^{0.5}(\lvert\Delta\rvert)$	Maximal exponent	$e^{\log^{0.5}(\lvert\Delta\rvert)}$
30	5	5	96
35	5	7	138
40	5	16	194
45	6	9	266
50	6	16	360
55	6	20	480
60	6	26	632
65	7	18	822
70	7	27	1060
75	7	35	1353
80	7	44	1714
85	8	38	2155
90	8	47	2693
95	8	58	3343
100	8	64	4128
105	9	60	5070
110	9	60	6198
115	9	83	7541
120	9	92	9138
125	9	120	11029
130	9	154	13261
135	10	107	15889
140	10	122	18976
145	10	145	22591
150	10	177	26814
155	10	228	31736
160	11	188	37462